06/01/04

AF/2124

EV444356732

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/665,214
Filing Date ..... September 18, 2000
Confirmation No.....4287
Inventorship..... ATM Shafiqul Khalid et al.
Applicant..... Microsoft Corporation
Group Art Unit 2124
Examiner ..... Chavis, John Q
Attorney's Docket No. MS1-571us
Title: *Access Redirector and Entry Reflector*

APPEAL BRIEF

To: MS: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

From: Kasey C. Christie (Tel. 509-324-9256; Fax 509-323-8979)
Customer No. 22801

Pursuant to 37 C.F.R. §1.192, Applicant hereby submits an appeal brief for Application No. 08/897,217. A Notice of Appeal was filed April 1, 2004. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

06/02/2004 MAHMED1 00000070 120769 09665214
01 FC:1402 330.00 DA

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

TABLE OF CONTENTS

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(1) Real Party in Interest

The real party in interest is the Microsoft Corporation, the assignee of all right and title to the subject invention.

(2) Related Appeals and Interferences

Appellant is not aware of any other appeals or interferences which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

(3) Status of Claims

Claims 1-40 stand rejected and are pending in this Application. No claims have been canceled and no claims have been allowed. Claims 13-18 and 31 have been Previously Presented and are set forth in the Appendix of Appealed Claims on page 23 with the remaining claims as originally presented or added.

Claims 1-2, 5-6, 8-9, 11-13, 15-16, 18-19, 21, 24-26, 28-35, and 37-39 stand rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,105,101 ("**Hester**"), as set forth in a Final Office Action dated October 8, 2003 (hereinafter, the "FINAL ACTION").

Claims 3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hester in view of U.S. Patent No. 5,655,148 ("**Richman**"), as set forth in the FINAL ACTION.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(4) Status of Amendments

The Applicant responded to the FINAL ACTION to address the 35 U.S.C. §102(a) rejections of pending claims 1-2, 5-6, 8-9, 11-13, 15-16, 18-19, 21, 24-26, 28-35, and 37-39 and the U.S.C. §103(a) rejections of pending claims 3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40.

Subsequently, an Advisory Action was issued on January 8, 2004 dismissing Applicant's traversal and maintaining the rejection of all pending claims. No other amendments have been filed subsequent to the Examiner's final rejection or ensuing Advisory Action.

(5) Summary of Invention

The present Application describes a technology that operates on a "common configuration data structure" to promote compatibility and interoperability between differing versions of program modules. The concepts of "configuration," "configuration databases," and "common configuration data structures" are discussed in the specification on pages 3-5. An example of a "common configuration data structure" is the "registry," which is discussed on page 5 of the specification.

A "common configuration data structure" is a data structure that includes configuration data, which describes how a software and/or hardware system is set-

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

up or configured, which is common to multiple program modules. Page 5, lines 18-23 says this:

> Herein, a common configuration data structure refers to a set of multiple configuration databases used by more than one version of a program module (such as an application). In addition, a common configuration data structure refers to a single configuration database (such as the Registry) used by more than one version of a program module (such as an application). A configuration database is often stored as one or more configuration files on the storage system of a computer.

When the technology described in the present Application receives an attempt (by a program module) to access one or more selected storage locations (i.e., "loci", "nodes") of the common configuration data structure, it interrupts such access (see blocks 200 and 202 of Fig. 2). Then it determines whether to redirect such attempted access to at least one other storage location of the same data structure (see block 204 of Fig. 2). Then it redirects the attempted access (see blocks 206-210 of Fig. 2).

The purpose of this redirection is to allow differing version of program modules to access their associated configuration information in the common configuration data structure without knowing it is doing so. (see the Abstract, "However, the differing versions are unaware that they are accessing different nodes.") This helps greatly for compatibility and of differing versions of program modules.

As shown in Fig. 3, as configuration information at a location is changed, the technology described in the present Application may copy selected portions of such changed information into its associated "reflected" location and vice versa.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

This reflection allows associated "reflected" locations to share relevant configuration information that promotes interoperability.

(6) Issues

A. Whether Hester anticipates claims 1-2, 5-6, 8-9, 11-13, 15-16, 18-19, 21, 24-26, 28-35, and 37-39 under 35 U.S.C. § 102(e) and the Office has satisfactorily met its burden to show such anticipation?

B. Whether the combination of Hester and Richman obviates claims 3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40, under 35 U.S.C. § 103(a) and the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper?

(7) Grouping of Claims

For each ground of rejection which applicant contests herein which applies to more than one claim, such additional claims, to the extent separately identified and augured below, do not stand or fall together.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(8) Argument

Issue A -- Whether Hester anticipates claims 1-2, 5-6, 8-9, 11-13, 15-16, 18-19, 21, 24-26, 28-35, and 37-39 under 35 U.S.C. § 102(e) and the Office has satisfactorily met its burden to show such anticipation?

In summary, Applicant submits that the Office has not satisfied its burden here because of the following reasons:

1. The BIOS, disclosed by Hester, is not a "common configuration data structure," as recited in all pending claims.
2. The BIOS interrupt calls, disclosed by Hester, are not an "attempt to access a storage locus," as recited in the claims 1-18, 29, 31, 33-36, and 39-40.
3. Hester does not disclose "intercepting an attempt by a program module to access configuration information ("config-info") of the program module," as recited in claims 12-20.
4. Hester does not disclose "a first storage locus containing configuration information ("config-info") for a first version of a program module; a second storage locus containing config-info for a second version of the program module," as recited in claim 29.
5. The Office has not met its burden to show anticipation with respect to claims 21-30, 32, and 37-38 by rejecting these claims on the grounds that their features are "inherent in claim 1 to enable backward compatibility...."

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Hester

Hester is the reference for the anticipation rejections and the primary reference for the later-discussed obviousness rejection. So, Applicant will initially and briefly focus on Hester here.

Hester describes a device driver that allows program modules designed to run one type of hardware arrangement to operate more effectively on a different (and presumably older) hardware arrangement.

More specifically, Hester describes its purpose as being a "mechanism of performing 16 Bit BIOS interrupt calls under a 32 Bit protected mode application" at col. 2, lines 16-17.

In other words, Hester allows for newer applications (e.g., a 32 Bit protected mode applications) to make its natural interrupt calls into the BIOS, but its interrupt calls are converted and/or redirected into a different type (e.g., 16 Bit), which is a older and backward-compatible type. That way, the BIOS of a hardware system need not be retrofitted in order to allow it to run the newer applications (e.g., 32 Bit protected mode). See Hester, col. 2 lines 12-33.

(1) The BIOS, disclosed by Hester, is not a "common configuration data structure," as recited in the claims 1-40 .

This group of claims (claims 1-40—which includes independent claims 1, 9, 12, 19, 21, 29-33, 37, and 39) recite a "common configuration data structure" feature. For example, claim 1 recites a method for "controlling access to storage loci in a common configuration data structure" and each of its two actions (e.g.,

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

"receiving" and "determining") operate, at least in part, on the "common configuration database."

If the meaning of a "common configuration data structure" is not sufficiently clear, then Applicant directs attention to the discussion of the concepts of configuration, configuration databases, and common configuration data structures in the specification beginning on page 3. That text is reproduced here:

> **Configuration**
>
> Configuration is the way a system is set up, or the assortment of components that make up the system. Configuration can refer to either hardware or software, or the combination of both. For instance, a typical configuration for a PC consists of 32MB (megabytes) main memory, a floppy drive, a hard disk, a modem, a CD-ROM drive, a VGA monitor, and an operating system.
>
> Many software products require that the computer have a certain minimum configuration. For example, the software might require a graphics display monitor and a video adapter, a particular microprocessor, and a minimum amount of main memory.
>
> When a person installs a new device or program, she sometimes needs to configure it, which means to set various switches and jumpers (for hardware) and to define values of parameters (for software). For example, the device or program may need to know what type of video adapter you have and what type of printer is connected to the computer. Thanks to new technologies, such as Plug-and-Play, much of this configuration is performed automatically.
>
> **Configuration Databases**
>
> Software applications typically employ one or more configuration databases to store configuration settings. Under some OSs (such as Windows® 3.1 and MS-DOS®), multiple configuration databases were used by the OS and the applications. There were files for starting the system (e.g., CONFIG.SYS and AUTOEXEC.BAT). There were files for connecting to a network (e.g., NETWORK.INI). There were files for running applications (e.g., WIN.INI and SYSTEM.INI).
>
> Each configuration file had its own rules and structure. Maintaining these files was a difficult chore for the OS. Providing a limited degree of synchronization between these files was also a difficult chore for the OS.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Common Configuration Data Structure

With the advent of more advanced OSs (such as Windows NT® and Windows® 95), a common configuration data structure was introduced. It is called the "Registry." All configuration settings are stored therein (except for other legacy configuration files that remained for backward compatibility reasons).

Herein, a common configuration data structure refers to a set of multiple configuration databases used by more than one version of a program module (such as an application). In addition, a common configuration data structure refers to a single configuration database (such as the Registry) used by more than one version of a program module (such as an application). A configuration database is often stored as one or more configuration files on the storage system of a computer.

The Registry. Certain OSs store and check the configuration information (herein, "config-info") at a single location—called the registry. Most applications write data to the registry, at least during installation. The registry is an example of a common configuration data structure.

The registry is a hierarchically structured database containing subtrees of keys that contain per-computer and per-user databases. Each key may contain data items called value entries and may contain subkeys. In the registry structure, keys (and subkeys) are analogous to directories and value entries are analogous to files.

The specification indicates twice (on p. 4, lines 15-16 and lines 21-22) that the "Registry" is an example of such a common configuration data structure. Furthermore, specification provides a definition of a "common configuration data structure" on p. 4, lines 18-22:

> Herein, a common configuration data structure refers to a set of multiple configuration databases used by more than one version of a program module (such as an application). In addition, a common configuration data structure refers to a single configuration database (such as the Registry) used by more than one version of a program module (such as an application).

The term "configuration" is defined in the specification on p. 3, lines 10-11: "Configuration is the way a system is set up, or the assortment of components that make up the system."

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

While term "database" is not expressly defined in the specification, Applicant submits that it is well-known to those of ordinary skill in the art. Applicant submits that it is an organized collection of data. More particularly, it is not typically known as set of computer-executable instructions or routines.

Webopedia™ (http://www.pcwebopaedia.com/) defines a "database" as follows [emphasis added]:

> "A collection of *information* organized in such a way that a computer program can quickly select desired pieces of *data*. You can think of a database as an electronic filing system.
>
> Traditional databases are organized by fields, records, and files. A field is a single piece of *information*; a record is one complete set of fields; and a file is a collection of records. For example, a telephone book is analogous to a file. It contains a list of records, each of which consists of three fields: name, address, and telephone number.

However, the specification does discuss the term "configuration database" on p. 4, lines 1-11. Specifically, lines 2-3 says, "Software applications typically employ one or more configuration databases to store configuration settings."

Applicant submits that the Office has not clearly identified what feature(s) in Hester that it equates to the "common configuration data structure" recited in the claims. Since the BIOS is the focus of Hester overall and the focus of nearly all of the portions of Hester cited by the Office, Applicant assumes that the Office equates the BIOS to the "common configuration data structure," as recited in the claims.

The BIOS, disclosed by Hester, is not a "common configuration data structure," as recited in the claims. Instead of providing configuration data as part

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

of a common configuration data structure, Hester's BIOS provides a set of basic instructions for controlling system hardware (Hester, col. 1, lines 44-49).

BIOS: Hester describes the BIOS as "The Basic Input/Output System (BIOS)" which is "typically coded into a computer system's ROM to provide the basic *instructions* for controlling system hardware. The operating system and application programs both directly access the BIOS *routines* to provide better compatibility for such functions as screen displays" [emphasis added] col. 1, lines 44-49.

Applicant emphasize the words "instructions" and "routines" here to point out the contrast with the definitions provided above from the specification and Webopedia™. Those definitions focus on data and information, which is the traditional and typical subject matter of databases.

Applicant submits that Hester's BIOS is not a "data structure." The BIOS of Hester contains instructions. The interrupt calls are for the BIOS to perform an action. If the BIOS disclosed by Hester happens to includes something that can be classified as data, Applicant submits that does not make the BIOS a database any more than novel is a cookbook because a fictional character bakes a cake.

Furthermore, Applicant submits that the BIOS is not a "configuration data structure." If the BIOS disclosed by Hester happens to includes something that can be classified as data, Applicant submits that is not sufficient. Rather, such data must be configuration data along the lines as that described in the specification on p. 3, lines 9-24.

Further still, Applicant submits that the BIOS is not a "common configuration data structure." If the BIOS disclosed by Hester happens to includes something that can be classified as configuration data, Applicant submits that is

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

not sufficient. Rather, such configuration data must be common so that it is "used by more than one version of a program module" (see Spec., p. 4, lines 18-22).

As indicated earlier, Applicant does not submit that it has invented a "common configuration data structure" (of which the "registry" is one example). Rather, Applicant submits that the claim language recited operates on a "common configuration data structure" and that Hester does not.

Accordingly, this group of claims (which includes all independent claims and their dependent claims; thus, all pending claims, where are claims 1-40) is allowable over the Hester for at least the reason that Hester does not disclose all of the recited features. In particular, Hester does not disclose the feature of a "common configuration data structure" and operations on such feature, as recited in these claims.

(2) The BIOS interrupt calls, disclosed by Hester, are not an "attempt to access a storage locus," as recited in the claims 1-18, 29, 31, 33-36, and 39-40.

This group of claims (claims 1-18, 29, 31, 33-36, and 39-40—which includes independent claims 1, 9, 12, 29, 31, 33, and 39) recite "accessing a storage locus" or something quite similar.

Applicant submits that the Hester does not disclose such. Furthermore, that the Office has not pointed out with particularity where Hester discloses such.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Applicant submits that, instead of "accessing a storage locus" (i.e., a storage location), Hester discloses making an interrupt call, which is a request to execute a set of instructions in the BIOS.

The BIOS interrupt calls, disclosed by Hester, are not an "attempt to access a storage locus," as recited in these claims. Instead, Hester's BIOS interrupt call is used to "get the attention" of some portion of a computer system (Hester, col. 1, lines 53-54) and execute instructions in the BIOS.

Interrupt Call: Hester says that "interrupt calls" are "used to '*get the attention*' of some portion of a computer system" [emphasis added] col. 1, lines 53-54. Furthermore, Webopedia™ (http://www.pcwebopaedia.com/) defines an interrupt call as follows [emphasis added]: "A signal informing a program that an event has occurred. When a program receives an interrupt signal, it takes a specified *action*."

Applicant submits that a signal intended to "get attention" or initiate an "action" does not equate to an "attempt to access a storage locus" (like that described in claim 1). An interrupt call does not attempt to read or write to particular storage locus.

Applicant respectfully submits that the Office cannot maintain the rejection of these claims without clearly pointing out, with particularity, where Hester discloses:

- a BIOS interrupt call that "access[es] a storage locus"; rather than acting as a request to execute a set of instructions in the BIOS.
- a redirection of such access to non-instructions (and more particularly configuration data) in the BIOS.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Accordingly, this group of claims (claims 1-18, 29, 31, 33-36, and 39-40) is allowable over the Hester for at least the reason that Hester does not disclose all of the recited features. In particular, Hester does not disclose the feature of an "attempt to access a storage locus," as recited in these claims.

(3) Hester does not disclose "intercepting an attempt by a program module to access configuration information ("config-info") of the program module," as recited in claims 12-20

This group of claims (claims 12-20—which includes independent claims 12 and 19) recite "intercepting an attempt by a program module to access configuration information ('config-info') of the program module."

Applicant submits that the Hester does not disclose such. Furthermore, that the Office has not pointed out with particularity where Hester discloses such.

Applicant submits that, instead of "intercepting an attempt by a program module to access configuration information ("config-info") of the program module," Hester discloses intercepting an interrupt call (Hester, col. 2, lines 14-33 and 48-65), which is a request to execute a set of instructions in the BIOS, and redirecting it to execute another set of instructions in the BIOS.

It is not, as the claims recite, an "attempt...to access configuration information ('config-info') of the program module." Instead, Hester attempts to initiate execution (rather than "access" data) of instructions. Those instructions are not, as the claims recite, "configuration information ('config-info') of the program module." Rather, they are, indeed, instructions.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Applicant respectfully submits that the Office cannot maintain its rejection of these claims without pointing out the specific portions of Hester that disclose configuration information *of the program module* and such information being stored the BIOS.

Accordingly, this group of claims (claims 12-20—which includes independent claims 12 and 19) is allowable over the Hester for at least the reason that Hester does not disclose all of the recited features. In particular, Hester does not disclose the feature of an "intercepting an attempt by a program module to access configuration information ("config-info") of the program module," as recited in these claims.

(4) Hester does not disclose "a first storage locus containing configuration information ("config-info") for a first version of a program module; a second storage locus containing config-info for a second version of the program module," as recited in claim 29

This claim (claim 29) recites "a first storage locus containing configuration information ("config-info") for a first version of a program module; a second storage locus containing config-info for a second version of the program module".

Applicant submits that the Hester does not disclose such. Furthermore, that the Office has not pointed out with particularity where Hester discloses such.

Applicant submits that, instead of "a first storage locus containing configuration information ("config-info") for a first version of a program module;

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

a second storage locus containing config-info for a second version of the program module," Hester discloses a BIOS containing sets of instructions which are executed when initiated by an interrupt call (Hester, col. 2, lines 14-33 and 48-65).

Applicant submits that Hester's BIOS contains no configuration information ("config-info"). Furthermore, Applicant submits that Hester's BIOS contains no config-info for differing versions of a program module at differing locations in the BIOS. Rather, Hester discloses a BIOS containing sets of instructions which are executed when initiated by an interrupt call

Applicant respectfully submits that the Office cannot maintain its rejection of this claim without pointing out the specific portions of Hester that disclose configuration information *of the first and second versions of program module* and such information being stored the BIOS.

Accordingly, this claim (claim 29) is allowable over the Hester for at least the reason that Hester does not disclose all of the recited features. In particular, Hester does not disclose the feature of an "a first storage locus containing configuration information ("config-info") for a first version of a program module; a second storage locus containing config-info for a second version of the program module," as recited in this claim.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(5) The Office has not met its burden to show anticipation with respect to claims 21-30, 32, and 37-38 by rejecting these claims on the grounds that their features are "inherent in claim 1 to enable backward compatibility...."

The Office has not met its burden to show anticipation with respect to this group of claims (claims 21-30, 32, and 37-38—which includes independent claims 21, 29, 30, 32, and 37) by rejecting these claims on the grounds that their features are "inherent in claim 1 to enable backward compatibility...."

Applicant respectfully submits that the Office has not met is burden and thus cannot maintain its rejection of these claims without "identify[ing] the elements of the claims, determin[ing] their meaning in light of the specification and prosecution history, and identify[ing] corresponding elements disclosed in the allegedly anticipating reference." Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 1458 (Fed. Cir. 1984).

For reference, Applicant notes the following from MPEP 2112, which instructs the Office on how to handle "inherency" in this situation:

> "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' " In re Robertson, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted)....
>
> "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

> flows from the teachings of the applied prior art." Ex parte Levy, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original)
> .....

The whole of the Office reasoning on this issue (from p. 3 of the Previous Office Action dated May 23, 2003 (hereinafter, the "PREVIOUS ACTION")) is quoted here:

> The features of claim[] 21 is inherent in claim 1 to enable backward compatibility; since, the intercepting and redirecting features inherently requires searching, and finding to acquire appropriate links and copying to enable calls to systems in different modes. The features of claim 24 are inherent in claim 21; since, a trigger event is inherent to enable interrupt calls.

Applicant submits that the Office has not provided a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of Hester.

Claim 1 recites a method for ***controlling access***. Acts of claim 1 include ***receiving*** an attempt to access a first storage locus and ***determining*** whether to direct such attempt to at least a second storage locus. On the other hand, claim 21 recites a method for ***replicating data***. Acts of claim 21 include ***searching*** multiple storage loci for modified data, ***finding*** modified data, and ***copying*** modified data.

There are no express acts of searching, finding, or copying within claim 1 and Applicant does not see where they exist implicitly. The method of claim 1 operates whether or not there is modified data. Claim 21, on the other hand, deals exclusively with the situation where modified data exists. Applicant respectfully submits that the features of claim 21 are not inherent in claim 1. If the Office maintains its rejection, Applicant respectfully requests clarification in the Office's arguments.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Accordingly, this group of claims (claims 21-30, 32, and 37-38—which includes independent claims 21, 29, 30, 32, and 37) is allowable over the Hester for at least the reason that the Office has not met its burden to show anticipation by Hester.

Issue B -- Whether the combination of Hester and Richman obviates claims 3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40, under 35 U.S.C. § 103(a) and the Office has satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper.

Each of these claims (3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40) which are rejected based upon obviousness ultimately depend from one of these independent claims: 1, 9, 12, 21, 29-33, 37, and 39. These independent claims stand rejected based upon anticipation, but, as Applicant indicated above, each of these independent claims is allowable.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

In addition to its own merits, each claim dependent from claims 1, 9, 12, 21, 29-33, 37, and 39 is allowable for the same reasons that its base claim is allowable.

Accordingly, Applicant submits that this claim grouping (claims 3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40) is also allowable over the Hester-Richman combination for at least the reason that the references do not teach or suggest the combination of claimed elements and features.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

Conclusion

Based upon the foregoing reasons, Applicant submits that Hester does not anticipate claims 1-2, 5-6, 8-9, 11-13, 15-16, 18-19, 21, 24-26, 28-35, and 37-39 under 35 U.S.C. § 102(e) and that the Office has not satisfactorily met its burden to show such anticipation. Also, based upon the foregoing reasons, Applicant submits that the combination of Hester and Richman does not obviates claims 3-4, 7, 10, 14, 17, 20, 22-23, 27, 36, and 40, under 35 U.S.C. § 103(a) and that the Office has not satisfactorily met its burden to show that these claims are obvious and that the combination of references is proper.

Applicant respectfully requests that the 35 U.S.C. §102(e) and the 35 U.S.C. §103(a) rejections be overturned and that pending claims 1-40 be allowed to issue.

Respectfully Submitted,

Dated: 5-28-04

By: _____

Kasey C. Christie
Reg. No. 40559
(509) 324-9256 x232
kasey@leehayes.com
www.leehayes.com

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

(9) Appendix of Appealed Claims

1. **(Original)** A method for controlling access to storage loci in a common configuration data structure, the method comprising:

receiving an attempt to access a first storage locus in the common configuration data structure from a program module;

determining whether to direct such attempt to at least a second locus in the common configuration data structure with the program module unaware that it is accessing the second locus.

2. **(Original)** A method as recited in claim 1 further comprising directing such attempt to at least the second locus, the program module being unaware that it is accessing the second locus.

3. **(Original)** A method as recited in claim 1 further comprising determining whether to direct such attempt to at least a third locus in the common configuration data structure with the program module is unaware that it is accessing the third locus.

4. **(Original)** A method as recited in claim 1 further comprising examining a loci-redirection table, wherein the determining is based, at least in part, upon information in the table.

5. **(Original)** A method as recited in claim 1, wherein the program module is an application.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

6. **(Original)** A method as recited in claim 1, wherein:

the first storage locus is reserved for configuration information ("config-info") for a first version of a program module;

the second storage locus is reserved for config-info for a second version of the program module.

7. **(Original)** A method as recited in claim 1, wherein the common configuration data structure is a registry.

8. **(Original)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 1.

9. **(Original)** A method for controlling access to storage loci in a common configuration data structure, the method comprising:

receiving an attempt to access a first storage locus in the common configuration data structure from a program module;

directing such attempt to at least a second locus in the common configuration data structure, the program module being unaware that it is accessing the second locus.

10. **(Original)** A method as recited in claim 9 further comprising directing such attempt to at least a third locus in the common configuration data structure, the program module being unaware that it is accessing the third locus.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

11. **(Original)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 9.

12. **(Original)** A method for directing an access to a storage locus in a common configuration data structure, the method comprising:

intercepting an attempt by a program module to access configuration information ("config-info") of the program module at a first storage locus in the common configuration data structure;

determining whether to redirect such attempt to at least a second locus in the common configuration data structure with the program module unaware that it is accessing its config-info at the second locus.

13. **(Previously Presented)** A method as recited in claim 12, further comprising redirecting such attempt to at least the second locus, the program module being unaware that it is accessing its config-info at the second locus.

14. **(Previously Presented)** A method as recited in claim 12, further comprising examining a loci-redirection table, wherein the determining is based, at least in part, upon information in the table.

15. **(Previously Presented)** A method as recited in claim 12, wherein the program module is an application.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

16. **(Previously Presented)** A method as recited in claim 12, wherein:

the first storage locus is reserved for configuration information ("config-info") for a first version of a program module;

the second storage locus is reserved for config-info for a second version of the program module.

17. **(Previously Presented)** A method as recited in claim 12, wherein the common configuration data structure is a registry.

18. **(Previously Presented)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 12.

19. **(Original)** A method for directing an access to a storage locus in a common configuration data structure, the method comprising:

intercepting an attempt by a program module to access configuration information ("config-info") of the program module at a first storage locus in the common configuration data structure;

redirecting such attempt to at least a second locus in the common configuration data structure, the program module being unaware that it is accessing its config-info at the second locus.

20. **(Original)** A method as recited in claim 19 further comprising redirecting such attempt to at least a third locus in the common configuration data structure, the program module being unaware that it is accessing the third locus.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

21. **(Original)** A method for replicating data in storage loci of a common configuration data structure of multiple storage loci, the method comprising:

searching multiple storage loci of the common configuration data structure for modified data;

finding modified data in a first storage locus;

copying selected modified data from the first storage locus to at least a second storage locus.

22. **(Original)** A method as recited in claim 21 further comprising copying selected modified data from the first storage locus to at least a third storage locus.

23. **(Original)** A method as recited in claim 21, wherein only storage loci listed in a loci-redirection table are searched during the searching.

24. **(Original)** A method comprising:

obtaining a triggering event that signals that a method as recited in claim 21 be initiated;

initiating such method as recited in claim 21.

25. **(Original)** A method as recited in claim 24 further comprising sending a triggering event when data in the common configuration data structure is altered.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

26. **(Original)** A method as recited in claim 21, wherein:

the first storage locus is reserved for configuration information ("config-info") for a first version of a program module;

the second storage locus is reserved for config-info for a second version of the program module.

27. **(Original)** A method as recited in claim 21, wherein the common configuration data structure is a registry.

28. **(Original)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 21.

29. **(Original)** A method of access redirection and entry reflection, the method comprising:

controlling access to storage loci in a common configuration data structure of multiple storage loci, the controlling comprising:

lee&hayes
421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com

- receiving an attempt to access a first storage locus in the common configuration data structure from a program module;
- directing such attempt to at least a second locus in the common configuration data structure, the program module being unaware that it is accessing the second locus;

replicating modified data in storage loci, the replicating comprising:

- searching multiple storage loci for modified data;
- finding modified data in at least one storage locus;
- copying selected modified data from the storage locus to at least another storage locus.

30. **(Original)** A computer-readable medium having computer-executable instructions that, when executed by a computer, perform a method for replicating data in storage loci of a common configuration data structure of multiple storage loci, the method comprising:

searching multiple storage loci of the common configuration data structure for modified data;

finding modified data in a first storage locus;

copying selected data from the first storage locus to at least a second storage locus.

31. **(Previously Presented)** An apparatus comprising:

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

a processor;

an access-redirector executable on the processor to:

receive an attempt to access a first storage locus in a common configuration data structure from a program module;

redirect such attempt to at least a second locus in the common configuration data structure, the program module being unaware that it is accessing the second locus.

32. **(Original)** An apparatus comprising:

a processor;

a entry-reflector executable on the processor to:

search multiple storage loci of a common configuration data structure for modified data;

find modified data in a first storage locus;

copy selected data from the first storage locus to at least a second storage locus.

33. **(Original)** An operating system comprising:

a common configuration data structure containing storage loci for storing configuration information ("config-info");

a loci-access redirector comprising:

receiver for receiving an attempt to access a first storage locus in the common configuration data structure from a program module;

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

director for directing such attempt to at least a second locus in the common configuration data structure, the program module being unaware that it is accessing the second locus.

34. **(Original)** An operating system as recited in claim 33, wherein the program module is an application.

35. **(Original)** An operating system as recited in claim 33, wherein:

the first storage locus is reserved for config-info for a first version of a program module;

the second storage locus is reserved for config-info for a second version of the program module.

36. **(Original)** An operating system as recited in claim 33, wherein the common configuration data structure is a registry.

37. **(Original)** An operating system comprising:

a common configuration data structure containing storage loci for storing configuration information ("config-info");

a loci-entry reflector comprising:

searcher for searching multiple storage loci of the common configuration data structure for modified data and for finding modified data in a first storage locus;

replicator for copying selected data from the first storage locus to at least a second storage locus.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes

38. **(Original)** An operating system as recited in claim 37, wherein:

the first storage locus is reserved for config-info for a first version of a program module;

the second storage locus is reserved for config-info for a second version of the program module.

39. **(Original)** A computer-readable medium having a common configuration data structure data structure, comprising:

a first storage locus containing configuration information ("config-info") for a first version of a program module;

a second storage locus containing config-info for a second version of the program module.

40. **(Original)** A computer-readable medium as recited in claim 39 further comprising a third storage locus containing a table that relates the first storage locus to the second storage locus.

421 West Riverside, Suite 500
Spokane, WA 99201
P: 509.324-9256
F: 509.323-8979
www.leehayes.com
lee&hayes